More Than a Feeling: A Lyrical Approach to Music Genre Classification

Final Exam Paper for Natural Language Processing and Text Analytics

Course Code: CDSCO1002E

Copenhagen Business School

MSc. Business Administration and Data Science

Student Name and Numbers:

Bellenberg, David	- 158373
Mealor, Alexander	- 158801
Ries, Alexander	- 158292
Torp, Aleksander	- 158277

Date of submission: 29.05.23

Number of Pages: 12

Number of characters: 33,915 characters

More Than a Feeling: A Lyrical Approach to Music Genre Classification

David Bellenberg
Alexander Mealor
Alexander Ries
Aleksander Torp
{dabe22af, alme22ab, alri22ac, alto22ad}@student.cbs.dk

Students - Copenhagen Business School
MSc. Business Administration and Data Science

May 29, 2023

Abstract

This paper investigates the applicability of machine learning for determining a song's genre based solely on lyrics. We implement five different combinations of models and word embedding techniques on a dataset of more than three million English songs spanning the genres pop, rap, rock, r/b, and country. The benchmark model is a Naive Bayes with Bag-of-Word vectorization which is compared against a Logistic Regression with TF-IDF vectorization, Random Forest and Long Short-Term Memory neural network with Word2Vec embeddings, and a BERT classifier. We find that the most complex model and embedding combination, BERT, provides the strongest classification performance in terms of macro average scores for both precision, 0.69, and recall, 0.61, across all genres. The results suggest that song lyrics may provide a solid foundation for classifying music genres. We propose an interactive user-interface incorporating our best performing model for real-time genre classification in the context of providers for music streaming.

***Keywords**: Lyrics, Text Classification, Naive Bayes, Logistic Regression, Random Forest, Word2Vec, LSTM, BERT*

1 Introduction

The number of songs within easy reach online has grown exponentially since the mid 2000's. On music streaming services, like Spotify, 100,000 new songs are uploaded daily (Ingham 2022) - handling such large amounts of data demands efficient music information retrieval systems and the automation of organization processes. Ordering songs according to their musical styles is convention, yet proves challenging due to a lack of standardized genre definitions - thereby introducing a subjective component - and the capacity of a song to fit multiple genres.

Moreover, both acoustic elements and lyrics, each with their own set of complexities add to the nuance. Hence, understanding the right characteristics of a song to extract for classification is an intricate endeavor. Research on genre classification has predominantly focused on investigating the properties of audio features, in conjunction with lyrical attributes. This is partly due to previous inaccessibility to large datasets of songs and lyrics and concerns over copyright issues, but more particularly because of the strong performance of classification models employing elements from both domains.

However, attaining and processing substantially large sets of data has since become feasible. With a dataset of more than three million song lyrics, this paper aims to advance the research field by utilizing and comparing the performance of Logistic Regression (LR), Random Forest (RF), Long Short-Term Memory Networks (LSTM), and Bidirectional Encoder Representations from Transformers (BERT) against a benchmark using Naive Bayes (NB) to classify songs into genres solely using lyrics. This will lay the foundation for an automated system enabling music classification by means of lyrical data, with the potential for music streaming services to streamline genre label recommendations for content uploaded.[1]

2 Related Work

Research on genre classification of music tends to emphasize using multiple features of a song, beside lyrics alone. Common practices involve combining textual properties of lyrics, such as stylistic, rhythmic and semantic features, and acoustic properties, including beat structures, pitch and volume levels, often in the form of time frequency representation images, as well as using multiple classifiers, correlation analysis, ensembling techniques and neural networks, and even some also considering genre classification in terms of cultural and mood aspects (Bassiou et al. 2015; Li et al. 2023; Mayer, Neumayer, et al. 2008a; Mayer and Rauber 2011; McKay et al. 2010; Neumayer and Rauber 2007; Yaslan and Cataltepe 2006; Ying et al. 2012).

Concerning work addressing genre classification purely based on lyrics, the scarce literature focuses primarily on composition of textual properties.

Mayer, Neumayer, et al. 2008b examine the usage of combinations of textual properties such as rhymes, stylistic features, and statistical and part-of-speech features of lyric. Their finding is that combining bag of words (BoW) methods with stylistic and rhythmic features of song texts increases the performance when using k-nearest neighbors, NB, Support vector machines and decision trees.

Howard et al. 2011 attempted to develop a system to handle lyrics classification regardless of language, comparing the use of BoW and NB, support vector classifier and decision trees. They found that the various classifiers perform differently across distinct languages and genres, and that removing stop words hurts performance, speculating that it removes informational cues.

Fell and Sporleder 2014 went deeper in lyric classification by analyzing multiple dimensions of a lyric's stylistic and linguistic features. They grouped features into vocabulary, style, semantics and orientation. The study experimented with how well these features could classify genres, identify ratings of music quality (i.e., best and worst), and publication time, and found that n-gram modeling can provide a reasonable approximation in these domains, but that combining the previously mentioned features enhances performance.

While much research concentrates on combinations of textual and audio features and various configurations of these, Ying et al. 2012, p. 260 points out that 'Lyrics associated with the music contain information that is vital to the reception and the message of a song' and, as

[1] The source code and data for this project are available. Source code: https://github.com/dbellenberg/GenrefromLyrics. Data: here.

Logan et al. 2004, p. 1 highlight ‘lyrics provide a much richer description of the song than simple forms of metadata such as the title, artist and year and arguably contain the true ‘content’ for many songs’. We therefore aim to examine the viability of determining the genre of a song by lyrics, as they are, alone. Moreover, since the research addressing only lyrics is both limited and relatively dated, and focuses on aggregations of textual properties of lyrics, we intend to examine the applicability of various NLP methods and models without dissecting the song texts.

Despite scant literature, the work of Boonyanit and Dahl 2021 comprise comparable research and inspiration for the purpose of this paper. They explored using GloVe word embeddings and LSTM models for genre classification based on lyrics on a dataset of 123,428 English songs with three genres: hip hop, pop and rock. After oversampling the classes to be roughly equal, they report a best performance of 0.68 accuracy using a LSTM model, against a baseline model using LR with an accuracy of 0.64.

3 Use Case

The ability to automate accurate genre classification is a task that large music streaming services must inevitably tackle as user demand and content produced for these platforms grows. Whilst users may be able to manually label song genres themselves, they are also prone to error and, critically, may have differing interpretations around the most appropriate genre label. Thus more objective, quantifiable measures of genre classification can help music streaming services better categorize large amounts of content.

Such categorization matters because it feeds into many key aspects of the business - personalized music recommendation systems, marketing efforts by artists, and industry trend analysis. Sub-genre categorisation or discovery may also begin with classification into broader categories before being further supplemented by methods such as topic modeling to identify new clusters.

Our task of song genre classification using lyrical content therefore has a clear use case in the form of its adoption on a large scale by a hypothetical music streaming service provider - who we refer to as ’BertBeats’. We envision an automated classification at point-of-publication which may offer a genre ‘recommendation’ to artists or record labels. Lyrical content is already available to the likes of Spotify, so its use in genre classification is realistic (Spotify 2021). Our *Related Work* discussion suggests that audio features tend to be given primacy in genre classification tasks currently, thus we look to investigate whether lyrical content alone can help to supplement existing methods.

4 Methodology

In this section, we outline the methodology applied for our research. First, we describe the dataset, followed by the exploration and general cleaning process. The data handling section ends with explaining the preprocessing measures for the lyrics, which are tailored according to the different models.

Furthermore, the machine learning approaches applied will be described. The complexity of the models is incrementally increased to test our hypotheses about performance of word embeddings and model complexity in genre classification. Finally, the section presents the evaluation metrics chosen for assessing model performance, justifying our focus on recall while also considering other performance metrics for a holistic evaluation.

4.1 Data Handling

The dataset for this research was acquired from Kaggle, comprising scraped song lyrics from Genius. By collecting and annotating song lyrics, Genius' community created a database that covers a wide array of music genres. The dataset contains over 5.9 million songs and spans across the music genres *Pop*, *R&B*, *Rock*, *Rap*, and *Country*, along with a *miscellaneous* category that contains songs from other genres not explicitly classified under the aforementioned genres.

The dataset comprises the 8 columns *title, tag, artist, year, views, features, lyrics, id*. The important columns for the analysis will be the *lyrics* and *tag* columns. The *lyrics* column contains the song lyrics, providing the input for genre classification. The *tag* column holds the genre tags that categorize each song into specific musical genres.

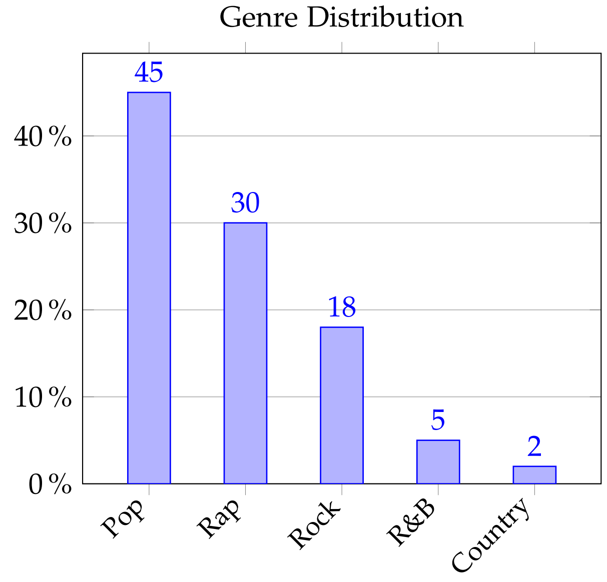


Figure 1: *Distribution of music genres in the dataset.*

As an initial step, we examined and cleaned the data for further steps of our research. Since the focus is on explicit genre classification, we removed all songs under the *miscellaneous* tag, refining our genre-specific dataset. Next, we observed that the dataset contained song lyrics in various languages. Given that our case is only interested in the classification of English song lyrics, we used FastText's language detection model, labelling all rows with language codes and filtering out songs that were not in English. Songs without a title or lyrics have also been removed from the dataset. This resulted in a reduced dataset of around 4.9 million rows.

Figure 1 illustrates a significant genre imbalance in the dataset. The smallest genre is *Country* with about 92,577 entries, while *Pop* includes over 1.8 million. This imbalance must be considered during the model training phase and the model evaluation.

Similarly, a wide variation is given in the years the songs were published. Therefore, we only included the songs released from 1960 to 2023, due to the following considerations:

- Language inherently evolves and changes over time, potentially complicating classification tasks.
- The majority of the songs in the dataset originated from the 2010s or later, suggesting that the impact of this decision would be minimal.
- The minor genres have largely grown in the recent years, as depicted in Figure 4 in the Appendix, which means that this cut-off also addresses the genre imbalance issue to some extent, excluding only songs from the genres *Pop*, *Rap* and *Rock*.

Furthermore, the exploration of the word count across different genres, illustrated in Figure 2, revealed that *Rap* songs, on average, contain more words than songs from other genres. The word count is essential for classifying the genre of a song from its lyrics. Therefore, we excluded we excluded songs with a word count of less than 25 and more than 5000 words. After performing the initial cleaning steps, the dataset contained 3,315,185 rows and was split into 70/15/15 train, validation, and test sets.

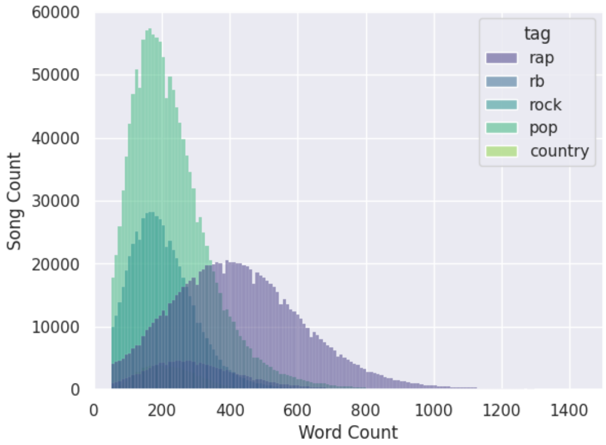


Figure 2: *Distribution of Total Number of Words per Song and Genre*

4.2 Lyrics Preprocessing

This process comprises several techniques aimed at cleaning and standardizing the lyrics, thereby ensuring its suitability for the different models. We removed characters which do not contribute to the meaning of the lyrics and may result in unnecessary noise during model training. Furthermore, we filtered out lyrical descriptions like "[Chorus]", since they provide no meaningful information for genre classification. Leading and trailing whitespaces were also removed.

Preprocessing	Models			
	NB	LR	W2V	BERT
Lowercasing	X	X	X	X
Tokenization	X	X	X	X
No Punctuation	X	X	X	
No Stopwords	X	X		

Table 1: *Preprocessing Techniques*

Besides these general steps, we applied different preprocessing measures based on the specific model being trained. These techniques were the removal of all punctuation, lowercasing, removal of stopwords and tokenization. Table 1 shows which of these techniques have been applied for each model.

4.3 Machine Learning Models

The approach to word embedding and model implementation taken here is an evolutionary one that develops incrementally. We choose to do so in order to test two hypotheses:

- *Hypothesis 1: Word embeddings of greater complexity will generally allow models to better classify music genres.*
- *Hypothesis 2: Models of greater complexity will exhibit superior genre classification performance.*

Naive Bayes and Logistic Regression
Naive Bayes (NB) and logistic regression (LR) are widely used methods in text classification, both because of their quick computations and capability of handling large and high-dimensional datasets (Thangaraj and Sivakami 2018). NB is also a recurrent baseline metric in related work, but considering the easiness and speed of implementation, LR is also a feasible and complementary metric. Consequently, we explore different configurations of NB and LR, vectorizing using either BoW or TF-IDF and oversampling and progress with the best performing. Table 3 in the Appendix displays the configurations and performance scores. Ultimately, the NB model with BoW embeddings serves as the benchmark to compare other models against. LR with TF-IDF is also adopted as a comparison model in order to test more sophisticated embedding techniques.

Word2Vec - Random Forest

In line with the comparison paper Boonyanit and Dahl 2021, we build upon the complexity of the word embeddings used in our analysis in relation to the baseline of BoW and TF-IDF. We hypothesize that more complex word embeddings themselves will lead to better classification performance by capturing greater semantic meaning from text. In using lyrical content - where subtle semantic relationships are frequent and context matters - we believe this embedding process is even more important. To do so, we train a Word2Vec (W2V) model on our corpus of song lyrics in order to generate more sophisticated embeddings. These capture word relations within the context of the corpus, unlike NB and LR which act as word counts. Preprocessing was similar to earlier methods, but stop words were retained in order to provide full context for model input - we note that this is more important for our second W2V implementation. Our training ignores words that feature less than 5 times in the corpus and sets a maximum window distance of 5 words between current and predicted words. Vectors of default dimensionality 100 and a continuous BoW architecture are used in order to limit computational complexity.

Although the comparison paper relies on GloVe embeddings, we generated W2V embeddings similar to their implementation in a second model, in which individual word vectors across a songs' lyrics are averaged to produce a single representative vector. 190 songs produce no vector and are thus dropped from the dataset. Applying dimension reduction via t-SNE and visualizing these average vectors in two dimensional space - as per Figure 3 - highlights the crossover in lyrical content observed between genres. Various levels of clustering do appear within *Rap* and *R&B* but separation between the likes of *Country* and *Rock* is less obvious. Visualizing the most frequently occurring 100-200 words across the vocabulary also highlights the relational aspect that W2V embeddings provided (see Figure 6 in the Appendix).

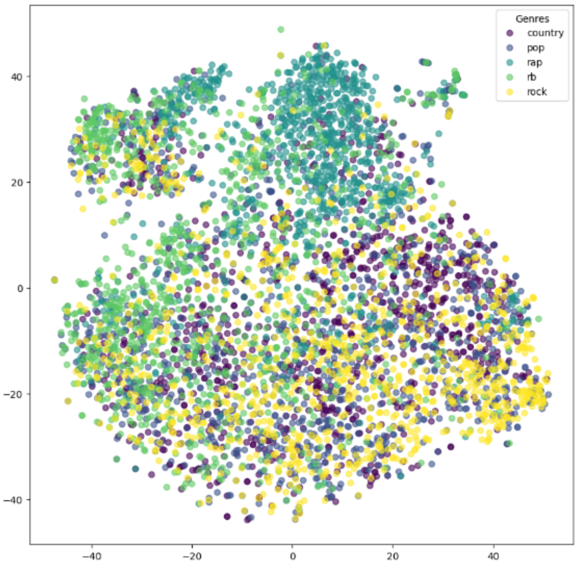


Figure 3: *Word2Vec Embeddings by Genre - Dimension Reduced via t-SNE*

For these embeddings we found the baseline NB model to be unsuited to handle their continuous nature. We therefore opted to use an RF classifier for its ability to handle the high dimensionality and continuity of W2V embeddings. We see similar approaches adopted by others in the literature with reasonable levels of success (Kumar et al. 2018). Our RF implementation consists of a default 100 number of estimators, as experimentation with higher numbers provided little performance gain for large computational losses.

Word2Vec - LSTM

While the Average Vector approach produces a single representative output for each song, much detail in its implementation is lost; word order within sequences and individual word meanings are foregone in the calculation of an average.

As Boonyanit and Dahl 2021 highlight, many of the same words feature across multiple genres. Word order may therefore add a further layer of identifiable information for classification. We therefore implement an embedding layer model which allows individual words to be fed, in order, into a classification model. To do so, lyrics are tokenized into indices, transformed into sequences and then uniformly padded or truncated into a vector of equal size.

We use the 75th percentile song length - 374 words - in order to capture the majority of songs whilst both limiting the degree of padding and level of computational load produced when running our already-large dataset. An embedding matrix consisting of all words in the corpus vocabulary is then produced with the embedding dimensions (100 in our W2V model) fitting the respective word vectors. We are then able to use this embedding matrix as the pre-trained weights in an embedding layer for a neural network.

To capture the informational value of word order, like the comparison paper, we implement a relatively simple LSTM model. It consists of the embedding layer, followed by a single LSTM layer of 128 units. Dropout and recurrent dropout of 0.2 are used to help regularize the model and prevent very early overfitting. The output dense layer uses a softmax activation function, outputting a probability distribution over the 5 genres. An Adam optimizer using a learning rate of 0.001 and categorical cross entropy loss were employed over 5 epochs and a batch size of 2048.

Underlying the decision to implement this more sophisticated approach - ultimately using the same form of embeddings - is our second hypothesis that greater model complexity will lead to superior genre classification performance.

BERT

Bidirectional Encoder Representations from Transformers (BERT) is a deep learning model designed for Natural Language Processing tasks. It is a transformer-based model, a state-of-the-art deep learning technique known for its efficiency in dealing with sequential data. Crucial to this is the attention mechanism that BERT utilizes to understand the context of the text from all directions (Devlin et al. 2019). This separates BERT from other models such as NB and LSTM, and allows it to consider the full context of a word by looking at the words that come before and after it.

BERT's architecture is composed of stacked transformer layers, each being a self-attention mechanism. In our study, we selected the BERT-base model considering its balance between performance and computational requirements. We preprocessed the lyrics similarly as with previous models but kept the stopwords because the model relies on the entire sentence structure for understanding the context.

The lyrics were tokenized and padded to a maximum length of 256 using the BertTokenizerFast. This breaking down of sentences into individual tokens ensures that all input sequences are of the same length, which is necessary for batching during model training.

We chose the 'bert-base-uncased' model, which treats uppercase and lowercase letters as the same, reducing the dimensionality of the input data. We adhered to the same train/validation/test split as the previous models but reduced the test set size due to BERT's longer inference time.

We set the batch size to 32 and used a learning rate of 1e-5, a value that has been shown to work well for text classification tasks with BERT. Given the computational expense associated with training a BERT model, coupled with the potential risk of overfitting, we limited the maximum number of epochs to 3. We saved the best model based on its performance on the validation set for our final testing phase.

4.4 Evaluation Metrics

A common first indicator of overall performance is accuracy. If the number of samples in each genre were equal, accuracy could have provided a good measure of the model's overall performance. However, our given data was very imbalanced among the different genres (see also Figure 1). Focusing only on accuracy as evaluation metric would be misleading, as a high accuracy reflects also the imbalanced class distribution. Furthermore, accuracy as evaluation metric works only well if the false positives and false negatives have similar cost.

When classifying genres from song lyrics on a large scale, predicting the maximum of the samples in one genre correctly is more important (recall) than having a high hit rate of correct classifications (precision). Therefore, we focused on recall as our primary metric. It is the ratio of true positive predictions to the sum of true positives and false negatives, calculated by the following formula:

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

Nevertheless, solely focusing on recall might have led us to more extreme measures such as undersampling of majority classes, oversampling of minority classes or adjusting class weights. While all methods would have increased the recall, they would also have led to an overfit of the minority classes or other undesirable side effects. For instance, oversampling makes the model overly sensitive to the minority class, causing an increase in false positives, meaning it will decrease also the precision and accuracy. Therefore, while recall was our primary concern due to the reasons discussed above, it is not our only consideration. We also looked at other metrics - precision and F1-score - to ensure that model performance was not only sensitive but also precise and balanced. By choosing these metrics with a focus on recall, we were able to gain a more comprehensive understanding of the model's performance and could adjust our strategies for model selection, training and tuning.

5 Results

A summary of the performance metrics for different machine learning models applied is shown in Table 2. All models appear to struggle in classifying certain genres and tend to have higher precision than recall, as shown in Table 4 in the Appendix. At first glance, precision and recall scores tend to be higher for the more present genres *Pop* and *Rap*. *Rap* has usually the highest recall in the models, whereas the recall rates for *R&B* and *Country* tend to be lowest.

The NB model, our baseline, has the lowest performance in terms of accuracy, and precision. It shows a good recall rate for *Pop* and *Rock* genres but struggles significantly with the rest. Unlike the other models, the baseline has a similar score between its macro average precision and recall. Furthermore, it is worth noting that the NB has the highest recall rates for genres *Rock* and *R&B*.

LR and RF models perform better than the baseline with regard to accuracy and precision. However, they display considerably low recall rates for *R&B*, *Rock*, and *Country* genres. The macro average recall for these models is worse than the baseline, as seen in Table 4 in the Appendix, resulting also in a worse F1-Score than the NB model. The LSTM model demonstrates slight improvement over LR and RF, particularly in terms of precision and accuracy. However, it also struggles with the recall rates for genres other than *Pop* and *Rap*.

The BERT model stands out as the best performer among the models tested. It has the highest F1 score, precision, and accuracy, and it shows the most balanced recall rate across the different genres. This balance can also be retrieved from the confusion matrix, presented in Figure 5 in the Appendix, underlining the highest recall rates for the genres *Country*, *Rap* and for the overall macro average. Also, the model has the highest precision of the *Pop* genre, whereas it shows the lowest recall rate of all models for this genre except the baseline. Overall, it's clear that BERT has delivered the most robust performance across metrics. Besides that, BERT is the only model that beats the macro average recall of our baseline model, as we see from Table 4 in the Appendix.

6 Discussion

Performance varied across models and genres, but we found that the BERT model provided the best results in terms of both precision and recall, affirming *Hypothesis 1* and *Hypothesis 2*, in that more complex word embeddings and models, particularly relative to our baseline, would lead to improved performance. However, since our evaluation of performance is layered, the outcome is more nuanced.

Firstly, the NB baseline - in relative terms - performs surprisingly well in slight contradiction to both of our hypotheses. Its poor performance in precision and accuracy is counterbalanced by its strong recall - outperforming all model and embedding combinations aside from BERT - and thus its F1-score. One suggestion as to why this may be is that lyric data is more sparse than anticipated. The approach of NB with BoW embeddings may better capture a large number of words that only appear in a limited number of songs. More sophisticated embedding approaches like those employed in W2V may not accurately capture the semantic meaning of such 'rare' words in the vocabulary. Still, BERT embeddings are able to capture a contextual element that W2V and BoW cannot, and thus the complexity here pays off in outperformance.

Secondly, the way in which embeddings are used clearly matters for performance. The underlying embedding data across our W2V implementations was identical, the averaging process utilized to allow its adoption in simpler models (RF) clearly leads to informational loss. Though computationally more expensive, the utilization of all individual word vectors in the embedding layer of a LSTM model lifted macro averages across all of the evaluation metrics.

Overall, both *Hypothesis 1* and *Hypothesis 2* as outlined appear generally true - in that BERT outperforms all previous combinations - but the path taken to reach it in our modelling implementations suggest that this is not straightforward.

Throughout all models, we faced challenges in classifying *Country* and *R&B* genres, which achieved relatively lower recall scores. This outcome could be due to the unique attributes of these genres. *Country* lyrics, for instance, might incorporate a wider range of vocabulary and themes, making them harder to distinguish. The lower scores for *R&B* could be attributed to its stylistic overlap with *Pop*, *Rap*, and *Rock*, which may confuse the model. Hence, these difficulties

Model	Precision	F1 Score	Accuracy	Recall				
	Macro Average			Pop	Rock	R&B	Rap	Country
Naive Bayes	0.49	0.47	0.59	0.49	0.62	0.35	0.79	0.18
Logistic Regression	0.61	0.46	0.69	0.85	0.29	0.08	0.85	0.11
Random Forest	0.63	0.40	0.68	0.88	0.21	0.03	0.84	0.04
LSTM	0.64	0.47	0.70	0.86	0.25	0.12	0.90	0.11
BERT	0.69	0.61	0.75	0.79	0.55	0.26	0.93	0.38

Table 2: *Summary of the performance metrics for various machine learning models used for music genre classification. Each model is evaluated based on Precision, F1 Score, Accuracy, and Recall for each of the five genres.*

hint at the inherent complexity in genre classification and its natural limitations.

The varying performance of the models across genres also prompts us to consider the influence of inter-genre variation. Given that genres often differ in linguistic complexity, vocabulary, and common themes, these discrepancies could be responsible for the differences in model performance. For instance, as illustrated in Figure 7 in the Appendix, *Rap*, characterized by its distinctive language pattern was generally easier for all models to distinguish from other genres.

6.1 Limitations

The skewed distribution of our dataset, favoring *Pop* and *Rap*, potentially influenced model performance. Our focus on song lyrics overlooked other genre-defining elements such as melody and rhythm. Also, the inherent subjectivity in genre classification suggests that even with a high-performing model, perfect classification is not feasible due to varying listener interpretations.

Concerning the quality of the scraped data, there could be drawbacks, which we were not able to control due to the sheer size of the dataset. We have no knowledge about the quality of the genre tagging process, measures of accuracy, and the potential for mislabelling, considering that Genius.com is a user-contribution platform.

6.2 Application & Potential Future Work

Our findings suggest that song lyrics alone may well be powerful enough features to classify songs by genre. However, this analysis covers only a limited number of genres, and even state of the art methods do struggle with lesser-represented categories. For research purposes, we propose that lyrical content be given greater representation in genre classification tasks. This analysis demonstrates that researchers need not only - or even primarily - rely on audio features in their classification tasks. Still, where lyrical overlap arises often - in our case, across the *Rap/R&B*, *Pop/Country* and *Rock/Pop* spheres - even the provision of context, word order and semantics that the likes of BERT handles may not be sufficient to efficiently demarcate genres. Thus, supplementing this analysis with audio features may help to better capture these nuances.

This same principle applies to our *Use Case* earlier outlined. Large music streaming platforms can readily implement the classification techniques we have proposed, either supplementing existing methods that rely mostly on audio features or starting from scratch. With the assistance of ChatGPT, we have developed a basic proposal UI [2], see Figure 8 in the Appendix for how a hypothetical music streaming service - 'BertBeats' - might integrate such models into their service pipelines. We implement our best performing model - BERT - into a user form at point of song upload onto the platform. Envisioning either music artists or record label representatives uploading content, when the 'Lyrics' field is filled and submitted, a suggested genre with likelihood percentages is provided based upon its content. As earlier discussed, such an objective system may help to better streamline music categorization by avoiding differing user interpretations or erroneous input. Handling classification at point of upload may also lead to more efficient pipelines into recommendation or marketing systems for users.

Future work in this space should consider expanding such classification to other languages and across a greater variety of genres and subgenres. Work with topic modelling techniques may help to better unveil new-found genres or linguistic themes that tie song clusters together. Additionally, treating genre classification as a binary relevance problem may help more effective training of models. By breaking up this multi-label problem into multiple independent binary tasks, as suggested by Zhang et al. 2018 – the difficulties that we found particularly between genres of high content crossover may become more manageable. Finally, our assumption that the dataset provided reflects a realistic distribution across genres is one that may need to be questioned in different contexts. Addition of further sub-genres may materially change the distribution and add to class imbalance. Further work should be done to handle this, as its impact on model performance is noticeable throughout most of our experimentation. Though oversampling techniques were tested throughout here, their implementation led to overly material trade-offs between precision and recall: more effective resampling methods may handle this better.

While our models rely purely on lyrical content, this does not exclude future approaches from integrating such classification models into existing systems based on audio features. Ensemble systems, the likes of which Mayer and Rauber 2011 propose, can be used to combine existing classifier outcomes based on audio features with those from a lyrical model.

7 Conclusion

In this paper we implemented five different combinations of model and word embedding technique in order to compare performance in music genre classification from song lyric content alone. This task was done with a view to addressing a use case of a large music streaming service improving its automated categorization of content. We hypothesized that more complex word embedding methods and classification models would lead to better performance in this task. Whilst both hypotheses were broadly correct - BERT embeddings and classification clearly outperformed our baseline NB model with Bag-of-Words vectorization - we did find that the simplest approach of our baseline performed surprisingly well relative to 'inbetween' testing models employing TF-IDF and averaged W2V embeddings, and more complex model implementations including RF. Our analysis suggests that the classification implementation chosen benefits greatly from the context that more

[2] The interactive UI is available at: https://huggingface.co/spaces/alexmealor/GenreFromLyrics

sophisticated embeddings can provide, which ultimately demands the use of more complex models. Utilizing our best trained BERT model, we demonstrated that such an implementation could readily be rolled out to fit the use case; a proposal user interface with live 'genre recommendation' for a music streaming service at point of song upload was developed. We suggest that such an addition - relying on lyrical content alone - is a worthy and realistic addition to a streaming service's categorization pipeline.

References

Bassiou, N., Kotropoulos, C., & Papazoglou-Chalikias, A. (2015). Greek folk music classification into two genres using lyrics and audio via canonical correlation analysis. *2015 9th International Symposium on Image and Signal Processing and Analysis (ISPA)*, 238–243. https://doi.org/10.1109/ISPA.2015.7306065

Boonyanit, A., & Dahl, A. (2021). Music genre classification using song lyrics. https://web.stanford.edu/class/cs224n/reports/final_reports/report003.pdf.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. https://doi.org/10.48550/arXiv.1810.04805

Fell, M., & Sporleder, C. (2014). Lyrics-based analysis and classification of music. *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 620–631. https://aclanthology.org/C14-1059

Howard, S., Silla Jr, C. N., & Johnson, C. G. (2011). Automatic lyrics-based music genre classification in a multilingual setting. *Proceedings of the Thirteenth Brazilian Symposium on Computer Music*.

Ingham, T. (2022). It's happened: 100,000 tracks are now being uploaded to streaming services like spotify each day. https://www.musicbusinessworldwide.com/its-happened-100000-tracks-are-now-being-uploaded/

Kumar, A., Rajpal, A., & Rathore, D. (2018). Genre classification using word embeddings and deep learning. *2018 International Conference on Advances in Computing, Communications and Informatics, ICACCI 2018*. https://doi.org/10.1109/ICACCI.2018.8554816

Li, Y., Zhang, Z., Ding, H., & Chang, L. (2023). Music genre classification based on fusing audio and lyric information. *Multimedia Tools and Applications*, *82*(13), 20157–20176. https://doi.org/10.1007/s11042-022-14252-6

Logan, B., Kositsky, A., & Moreno, P. (2004). Semantic analysis of song lyrics. *2004 IEEE International Conference on Multimedia and Expo (ICME) (IEEE Cat. No.04TH8763)*, *2*, 827–830 Vol.2. https://doi.org/10.1109/ICME.2004.1394328

Mayer, R., Neumayer, R., & Rauber, A. (2008a). Combination of audio and lyrics features for genre classification in digital audio collections. *Proceedings of the 16th ACM International Conference on Multimedia*. https://doi.org/10.1145/1459359.1459382

Mayer, R., Neumayer, R., & Rauber, A. (2008b). Rhyme and style features for musical genre classification by song lyrics. *Ismir*, *14*(18), 337–342.

Mayer, R., & Rauber, A. (2011). Musical genre classification by ensembles of audio and lyrics features. *Proceedings of International*

Conference on Music Information Retrieval, 675–680. https://ismir2011.ismir.net/papers/PS6-4.pdf.

McKay, C., Burgoyne, J., Hockman, J., Smith, J., Vigliensoni, G., & Fujinaga, I. (2010). Evaluating the genre classification performance of lyrical features relative to audio, symbolic and cultural features. *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010*, 213–218. https://archives.ismir.net/ismir2010/paper/000038.pdf.

Neumayer, R., & Rauber, A. (2007). Integration of text and audio features for genre classification in music information retrieval. *Lecture Notes in Computer Science*, 724–727. https://doi.org/10.1007/978-3-540-71496-5_78

Spotify. (2021). You can now find the lyrics to your favorite songs in spotify. here's how. https://newsroom.spotify.com/2021-11-18/you-can-now-find-the-lyrics-to-your-favorite-songs-in-spotify-heres-how/

Thangaraj, M., & Sivakami, M. (2018). Text classification techniques: A literature review. *Interdisciplinary Journal of Information, Knowledge, and Management*, *13*. https://doi.org/10.28945/4066

Yaslan, Y., & Cataltepe, Z. (2006). Audio music genre classification using different classifiers and feature selection methods. *18th International Conference on Pattern Recognition (ICPR'06)*, *2*, 573–576. https://doi.org/10.1109/ICPR.2006.282

Ying, T. C., Doraisamy, S., & Abdullah, L. N. (2012). Genre and mood classification using lyric features. *2012 International Conference on Information Retrieval Knowledge Management*, 260–263. https://doi.org/10.1109/InfRKM.2012.6204985

Zhang, M.-L., Li, Y.-K., Liu, X.-Y., & Geng, X. (2018). Binary relevance for multi-label learning: An overview. *Frontiers of Computer Science*, *12*(2), 191–202. https://doi.org/10.1007/s11704-017-7031-7

Appendix

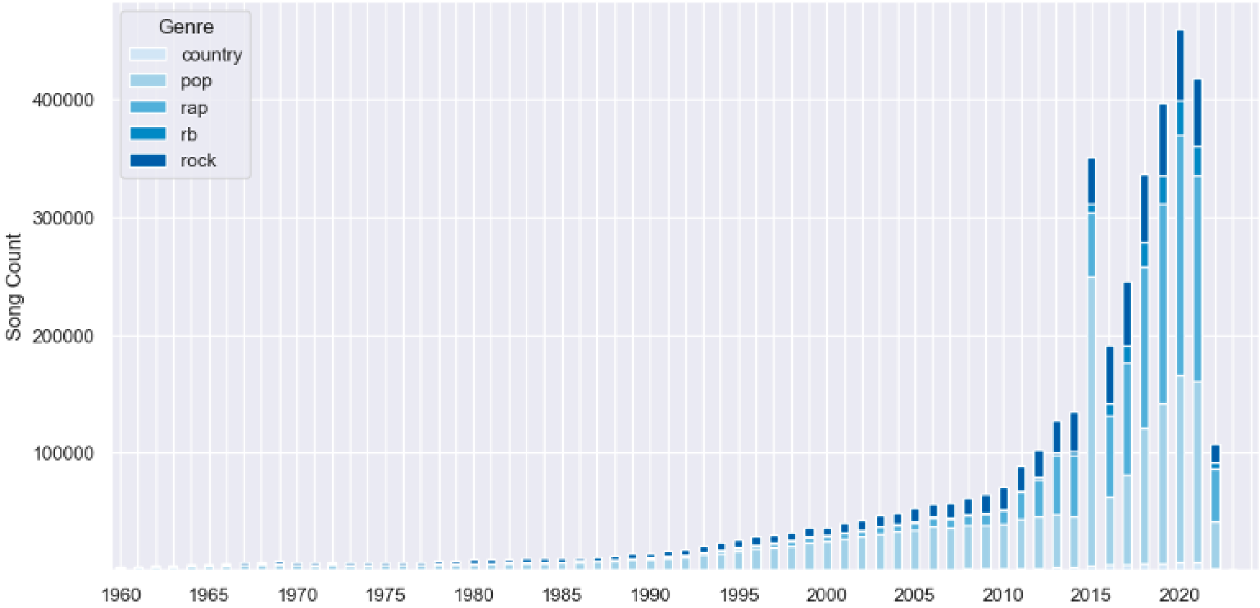


Figure 4: *Yearly Genre Distribution from 1960 to 2022*

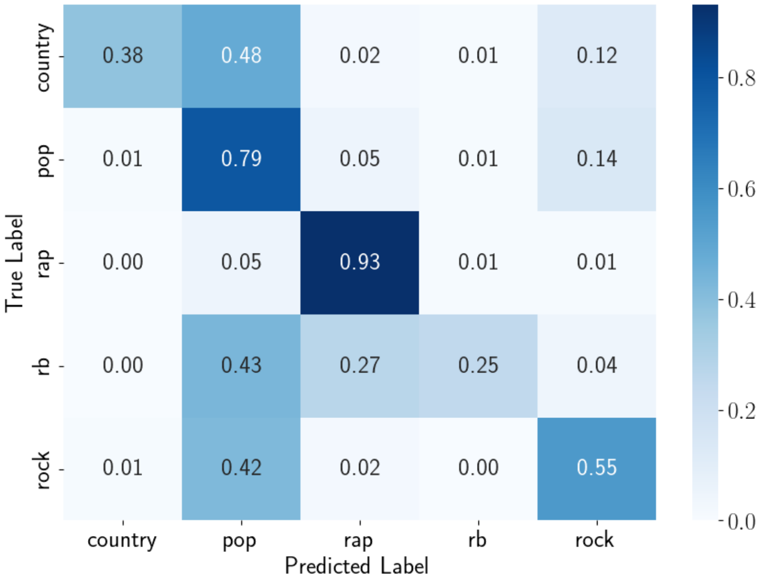


Figure 5: *Confusion Matrix of BERT in Percentage (%)*

Data	Classifier & Configures	Precision	Recall	F1	Time
Tokenized	NB BOW	0.49	0.49	0.47	7
	NB TF-IDF	0.45	0.36	0.32	7
	NB BOW ROS	0.45	0.57	0.41	7
	LR BOW	0.58	0.43	0.45	17
	LR TF-IDF	0.61	0.44	0.46	17
	LR TF-IDF ROS	0.47	0.60	0.46	25
Stemmed	NB BOW	0.49	0.49	0.47	6
	NB TF-IDF	0.44	0.36	0.32	7
	NB BOW ROS	0.45	0.57	0.41	7
	LR BOW	0.59	0.42	0.44	17
	LR TF-IDF	0.60	0.43	0.45	17
	LR TF-IDF ROS	0.47	0.60	0.46	22
Lemmatized	NB BOW	0.49	0.48	0.47	6
	NB TF-IDF	0.44	0.36	0.32	6
	NB BOW ROS	0.45	0.57	0.41	7
	LR BOW	0.59	0.43	0.44	16
	LR TF-IDF	0.61	0.44	0.46	15
	LR TF-IDF ROS	0.47	0.60	0.46	22

Table 3: *Performance comparison of different classifiers and configurations on tokenized, stemmed, and lemmatized lyrics. Here, "NB" stands for Naive Bayes, "LR" represents Logistic Regression, "BOW" refers to Bag of Words, "TF-IDF" denotes Term Frequency-Inverse Document Frequency, and "ROS" is an acronym for Random Over Sampling. Time in minutes.*

Model	Label	Precision	Recall	F1-Score
Naive Bayes	country	0.33	0.18	0.23
	pop	0.65	0.49	0.56
	rap	0.83	0.79	0.81
	rb	0.25	0.35	0.29
	rock	0.39	0.62	0.48
	Macro Avg	0.49	0.49	0.47
	Accuracy	0.59		
Logistic Regression	country	0.52	0.11	0.18
	pop	0.62	0.85	0.72
	rap	0.86	0.85	0.85
	rb	0.47	0.08	0.14
	rock	0.57	0.29	0.38
	Macro Avg	0.61	0.44	0.46
	Accuracy	0.69		
Random Forest	country	0.55	0.04	0.07
	pop	0.60	0.88	0.72
	rap	0.85	0.84	0.85
	rb	0.55	0.03	0.06
	rock	0.59	0.21	0.30
	Macro Avg	0.63	0.40	0.40
	Accuracy	0.68		
LSTM	country	0.63	0.11	0.19
	pop	0.64	0.86	0.73
	rap	0.85	0.90	0.87
	rb	0.47	0.12	0.19
	rock	0.62	0.25	0.36
	Macro Avg	0.64	0.45	0.47
	Accuracy	0.70		
BERT	country	0.66	0.38	0.48
	pop	0.72	0.79	0.76
	rap	0.88	0.93	0.90
	rb	0.57	0.26	0.36
	rock	0.61	0.55	0.58
	Macro Avg	0.69	0.58	0.61
	Accuracy	0.75		

Table 4: *Comparison of text classification models for different music genres.*

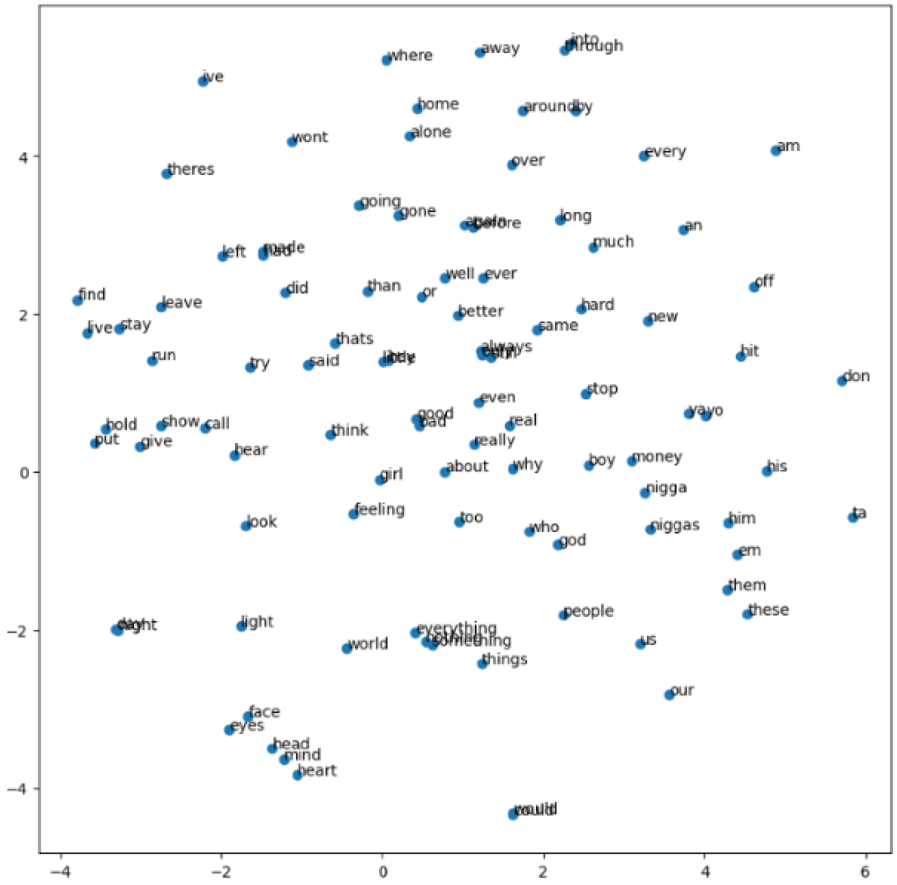
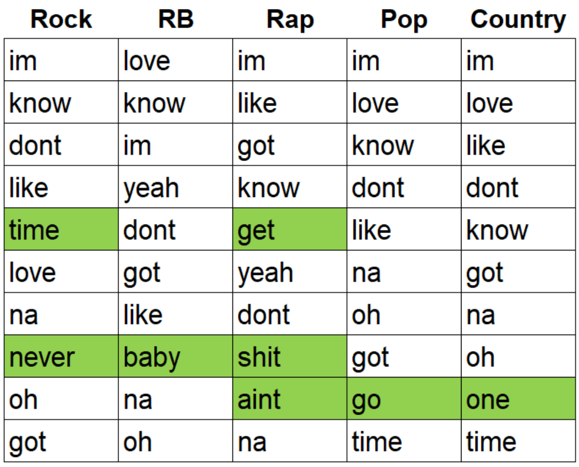
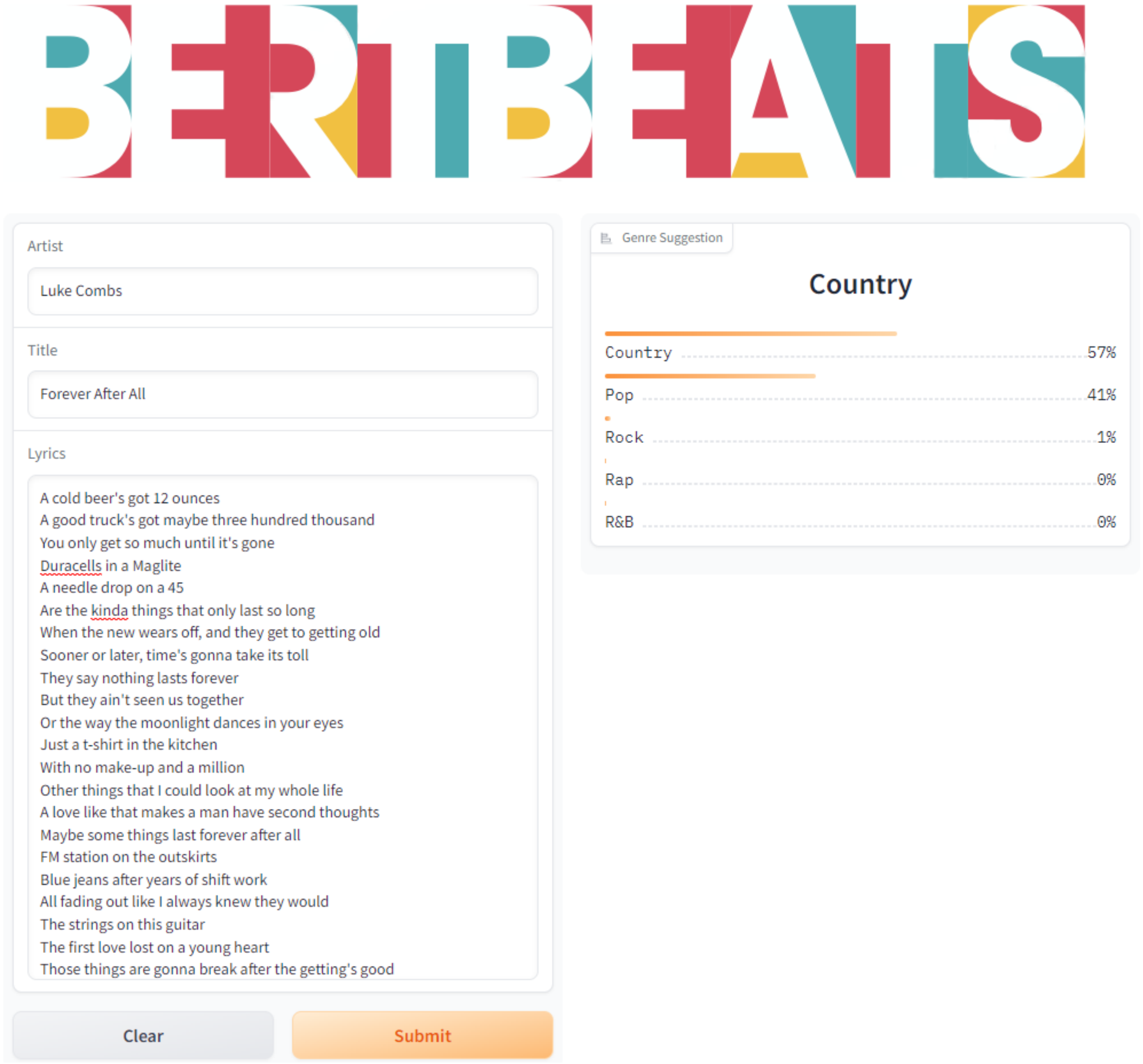


Figure 6: *100-200 Most Frequently Occurring Words - Word2Vec Embeddings*



Rock	RB	Rap	Pop	Country
im	love	im	im	im
know	know	like	love	love
dont	im	got	know	like
like	yeah	know	dont	dont
time	dont	get	like	know
love	got	yeah	na	got
na	like	dont	oh	na
never	baby	shit	got	oh
oh	na	aint	go	one
got	oh	na	time	time

Figure 7: *Most Important Features per Genre, highlighted words are unique within the table. Feature extracted using the Bag-of-Words approach for the NB Classifier*



Artist

Luke Combs

Title

Forever After All

Lyrics

A cold beer's got 12 ounces
A good truck's got maybe three hundred thousand
You only get so much until it's gone
Duracells in a Maglite
A needle drop on a 45
Are the kinda things that only last so long
When the new wears off, and they get to getting old
Sooner or later, time's gonna take its toll
They say nothing lasts forever
But they ain't seen us together
Or the way the moonlight dances in your eyes
Just a t-shirt in the kitchen
With no make-up and a million
Other things that I could look at my whole life
A love like that makes a man have second thoughts
Maybe some things last forever after all
FM station on the outskirts
Blue jeans after years of shift work
All fading out like I always knew they would
The strings on this guitar
The first love lost on a young heart
Those things are gonna break after the getting's good

Genre Suggestion

Country

Country 57%
Pop 41%
Rock 1%
Rap 0%
R&B 0%

Figure 8: *Example User Interface with Recommended Genre - available at https://huggingface.co/spaces/alexmealor/GenreFromLyrics*